To appear at SPIE Conf. on Telemanipulator and Telepresence Technologies VI, (SPIE 3840), Boston MA, Sep., 1999.

Combinations of Non-rigid Deformable Appearance Models

Saratendu Sethi and Stan Sclaroff

Image and Video Computing Group
Computer Science Department
Boston University
Boston, MA 02215, USA.

ABSTRACT

A framework for object recognition via combinations of nonrigid deformable appearance models is described. An object category is represented as a combination of deformed prototypical images. An object in an image can be represented in terms of its geometry (shape) and its texture (visual appearance). We employ finite element based methods to represent the shape deformations more reliably and automatically register the object images by warping them onto the underlying finite element mesh for each prototype shape. Vectors of objects from the same class (like faces) can be thought to define an object subspace. Assuming that we have enough prototype images that encompass major variations inside the class, we can span the complete object subspace. Thereafter, by virtue of our subspace assumption, we can express any novel object from the same class as a combination of the prototype vectors. We present experimental results to evaluate this strategy and finally, explore the usefulness of the combination parameters for analysis, recognition and low-dimensional object encoding.

Keywords: Deformable objects, finite element models, appearance models, linear combinations.

1. INTRODUCTION

Optimal and reliable description of objects has been one of the primary goals of *computer vision*. Significant amount of research has been conducted for deriving mathematical models of objects from images. Such descriptions have been useful for purposes like object recognition and image analysis. Important characteristics of such descriptions are that they should be easily computable and unique.

A common strategy employed in computer vision to design effective algorithms is to emulate methods of reasoning believed to be used by human beings to perform *image analysis*. Various psychophysical and physiological studies[1–6] have indicated that the human visual system uses strategies that encode three dimensional objects as multiple viewpoint-specific representations that are largely two-dimensional with appropriate depth information.[3,5] Various test evidences and computational simulations indicate that view interpolation offers a plausible explanation for viewpoint dependent performance of human response times and error rates for recognition.[1]

The psychophysical studies stated above strongly motivate us to devise algorithms that will represent object classes in terms of 2D prototype images. An object can be fully described by its two components, namely shape and appearance*. Hence, given sufficient number of good prototypes that encompass appropriate in-class variations, our goal is to build a deformable appearance model that will reliably describe an object class by linear combination of prototypes, i.e. the parameters of a novel object can be obtained by a linear combination of the prototype parameters in the training set.

Figure 1 gives a pictorial description of the approach for three prototype images. The images at the vertices of the triangle represent three prototypes. The prototypes can be registered with each other by warping them onto the average image. The shape and texture of each prototype can then be combined to generate new images. Here the intermediate images are represented along the edges of the triangle. For each intermediate image, the contribution of the adjacent prototypes is more than that of the one farther away.

*The notion of appearance is same as the object texture. Both terms will be used interchangeably

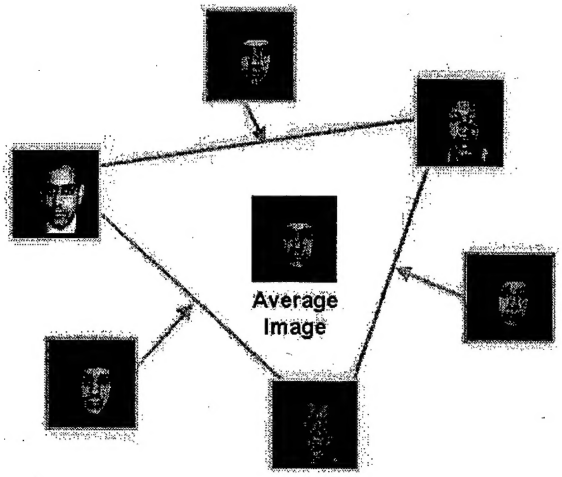

Figure 1. Basic Idea: Combination of prototypes. The images at the vertices of the triangle are the prototype images which can be registered with each other by warping onto the average image. The prototype shape and the texture parameters obtained by registration can be combined to generate new images.

2. OBJECT REPRESENTATION

Based on the premise that objects can be represented by their shape and appearance, computer vision algorithms for object representation can be categorized into two classes, namely *shape models* and *appearance models*. Initial techniques for shape and appearance modeling, were built independent of each other. Either class of techniques, ignored the parameterization of the other feature. This section throws light on some deformable shape and view-based representations relevant for the formulation of our approach and then explores some methods that try to combine both representations to handle greater variations robustly.

2.1. Shape Modeling

Initial shape representation methods concentrated on ways to employ flexible models by constraining the solution space of allowed deformations. Kass, Witkin and Terzopoulos[7] described a method of representing objects in images as *active contours* or energy minimizing splines that were guided by external constraint forces and influenced by image forces along the image gradients. Cootes proposed the *Chord Length Distribution* or the *Point Distribution Model*,[8] a method of shape representation that estimates the chord-lengths where each object is represented as an n-vertex polygon.

Sclaroff and Pentland[9,10] had proposed a method of representing objects in terms of *modal descriptions*, which is based on the idea of describing objects by their generalized symmetries, as defined by the object's deformation modes. Unlike the *point distribution model*, which statistically modeled object shapes, this method physically modeled objects by determining the modes of free vibrations of the object. The modes of an object define an orthogonal object-centered coordinate system where each feature point can be uniquely described as a combination of those modes. Cootes and Taylor later proposed a new method by combining this physically based method with their statistically based method.[11]

2.2. Appearance Modeling

Appearance-based models seek to obtain a compact representation for intensity distribution. One such set of techniques employ eigen-based methods to compress an image by projecting it onto a low-dimensional orthogonal basis, the *eigenspace*.[12–15] This orthogonal basis is usually statistically learned by using *principal components analysis* (PCA) or Karhunen-Loeve expansion on a large set of training data.

The concept of point distribution models was extended to model intensity distributions. These models are known as *Appearance models*[16] and have been claimed to address the problem of shape normalization which was not addressed in eigenfaces. This method requires labeled examples for training.

2.3. Combined Shape and Appearance Modeling

In order to avoid the implicit parameterization of shape in appearance models and make shape models more photo-realistic, there has been a growing interest in modeling both shape and appearance in a single model. Nastar, Moghaddam and Pentland[17] had combined physically based modes of vibrations with statistically-based modes of variation by considering each point in the image as a triplet of $(x, y, I(x, y))$ and doing manifold matching in this XYI space. Although this method combined both the statistical and physical modes of variation, it is dependent on good initialization.

Ullman and Basri[18] have showed that an object can be represented as a combination of 2-D images where the images are represented in terms of some linear transformations in the 3-D space. However, this method assumes a linear framework for object deformations and handles only limited non-rigid deformations.

Poggio, Jones and Vetter[19–22] have suggested that given sufficient number of prototypes, the parameter vectors define a linear space and span the model space. Any novel object can then be expressed as some combination of those prototype vectors. This method combines shape or geometry with texture or appearance in a way that minimizes both shape and appearance parameters to fit the model. This is a robust method as the problem of model fitting is solved as a global non-linear minimization problem.

Cootes, Edwards and Taylor[23] have also suggested a combined formulation of their appearance and active shape models to develop a new model known as the *Active Appearance Models*. This method does PCA in both the shape and the texture spaces separately and then combines them and again does PCA to remove redundancies between shape and texture parameters. All objects are then represented as some combination in this orthogonal model.

3. MATHEMATICAL FORMULATION

Let $I_1, I_2, \ldots, I_N$ be the N prototypes available for training the system. Let I_{ref} be the reference image. The objective is to define a framework whereby all the prototype images can be combined to generate images of novel objects from the same class. The formulation described here is similar in flavor to that developed by Jones and Poggio,[19] though the shape deformations are determined by finite element methods as opposed to optical flow methods. The prototype images are initially not in correspondence and hence cannot be combined. This emphasizes the determination of pixel to pixel correspondences amongst the prototype images. Let $S_1, S_2, \ldots, S_N$ be a set of shape parameters such that each S_i can be used to warp the *ith* prototype image onto the reference image, thereby bringing the prototype image into correspondence with the reference image, i.e.

$$S_i(x, y) = (\hat{x}, \hat{y}) \tag{1}$$

where $(\hat{x}, \hat{y})$ is the point in I_i which corresponds to (x, y) in I_{ref}. We define,

$$T_i(x, y) = \mathcal{W}^{-1}(I_i, S_i)(x, y) \tag{2}$$

where $\mathcal{W}$ is the warping function. Thus, for each prototype I_i in the training set, we obtain a shape vector S_i and a inverse warped texture vector T_i. Note that the texture vectors are shape-free as all of the prototype images are inverse warped onto the same reference prototype image.

Given a large number of prototypes which appropriately vary from each other with respect to different characteristics of the object class, we can define a set of parameters $\mathbf{b} = [b_1, b_2, \ldots, b_N]$ and $\mathbf{c} = [c_1, c_2, \ldots, c_N]$ such that the shape and the texture of a novel object I_{novel} (not in the prototype set) can be derived as a combination of the prototype shape and texture parameters.

$$S_{novel} = \sum_{i=1}^{N} c_i S_i = \mathbf{c} \cdot \mathbf{S} \tag{3}$$

$$T_{novel} = \sum_{i=1}^{N} b_i T_i = \mathbf{b} \cdot \mathbf{T} \tag{4}$$

Therefore, the equation for the novel image can be defined as follows:

$$\mathcal{W}^{-1}(I_{novel}, \mathbf{c} \cdot \mathbf{S}) = \mathbf{b} \cdot \mathbf{T} \tag{5}$$

Hence the matching phase reduces to matching the the novel image, which can be done by minimizing the sum of squared differences (SSD) error

$$E(\mathbf{c},\mathbf{b}) = \frac{1}{2}\sum_{\mathbf{x},\mathbf{y}}[\mathcal{W}^{-1}(\mathbf{I_{novel}}, \mathbf{c}\cdot\mathbf{S})(\mathbf{x},\mathbf{y}) - (\mathbf{b}\cdot\mathbf{T})(\mathbf{x},\mathbf{y})]^2 \tag{6}$$

The values of the parameters **c** and **b** so obtained, provide a compact representation of the novel image in terms of the prototypes in the training set. Since, the shape and the texture vectors of the prototypes define two completely different linear subspaces for the object class and may or may not be independent of each other, an important caveat involved here is the combined estimation of both the shape and texture parameters. Equation 6 is the basic equation that describes the mathematical formulation of the system. Further constraints may be employed depending upon the modeling of the parameters (see Section 5). We use a non-linear technique for minimization. In order to avoid getting trapped in the local minima, we use *Gaussian pyramids.* Both topics are described in brief here.

3.1. Minimization

For the minimization of the objective function, we use *Levenberg-Marquardt method,*[24] a non-linear optimization technique. This technique uses a combination of linear and non-linear approaches for updating parameters during each iteration. Smooth switching between the two approaches is accomplished by a weighting term λ. When the magnitude of λ is low, the minimization is done in a linearized fashion by *Gauss-Newton method* whereas higher magnitude of λ forces the system to be solved in quadratic fashion by using *Gradient Descent technique.*

The mathematical formulation is as follows. Given an objective function E, the parameters of which are q', the goal is to determine an instance q that minimizes the value of E. This is achieved iteratively by solving the following set of simultaneous equations:

$$(H + \lambda I)\Delta q = g \tag{7}$$

$$q' = q + \Delta q \tag{8}$$

where H, g and λ are the Hessian matrix, the gradient vector and the controlling parameter respectively. The gradient vector and the Hessian matrix are determined as follows:

$$g_k = -\frac{\partial E}{\partial q_k} \tag{9}$$

$$h_{kl} = \frac{\partial E}{\partial q_k}\frac{\partial E}{\partial q_l} \tag{10}$$

The cost of the objective function is determined with the updated parameter values q'. If the cost has decreased as compared to its previous value then the system tends to linear minimization by scaling down λ by a factor of 10. If the cost has increased then the system moves towards quadratic minimization by scaling up λ by 10. In the former case, the parameters are updated to q', whereas in the latter case, the updated parameter vector q' is discarded and we proceed with the old parameter vector q. Higher values of λ restrict parameter displacement in the error space and force the solution to move along the *steepest gradient.* Equations for computing various derivatives mentioned here will be provided in Section 5.

3.2. Gaussian Pyramids

It is not uncommon to find situations where the minimization solution gets trapped in local minima. This may happen when the error function is not exactly concave or the amount of change allowed in the parameters do not move the current estimate closer to the global minima. As a result the solution gradually drifts into a local trough and eventually gets trapped inside there. Such problems can be handled reliably by using a *multigrid relaxation approach.*[25] These methods work by taking advantage of multiple discretizations and smoothing of a continuous problem over a range of resolution levels. Solution to a minimization problem requires computations proportional to the spatial distance between the current estimate and the actual solution. This suggests the possibility of speedup by computing the solution over a coarse grid and then enhance it by successively refining the grid. Pyramids are one such multi-resolution technique used in image processing.[26]

The pyramids used in our implementation are called *octave pyramids* as at each level the image is halved in each dimension and subsampled. Successive reduction in the resolution and subsampling results in the loss of high

frequency components in the original image. In other words, this is equivalent to filtering the image through low-pass filters whereby the image is blurred by Gaussian kernels at each level. Thus at the coarsest level, it may be assumed that all the components corresponding to the local minima are smoothened enough to be determined as possible points of solution. Hence when successive solutions are computed from the coarsest levels and propagated to the finer levels, the solution tends towards the global minima and eventually it may be expected to converge to the actual global minima.

4. DEFORMABLE SHAPE MODELING

Shape modeling in our system is done by using *finite element models* (FEM).[10,27] The advantage of finite element models is their ability to enforce *a priori* constraints on smoothness and amount of deformation, which in general is not possible in statistically based or optical flow based methods. FEM is a numerical approach for *modal analysis* which can be used for describing non-rigid deformations of an elastic body. In this formulation, an object is modeled as a sheet of rubber which can freely deform. The surface of the object is interpolated by Galerkin method.[28] A set of polynomial functions are defined that relate the displacement of a single point to the relative displacements of other points. Hence all the points can be expressed in terms of the interpolation functions as below:

$$\mathbf{u(x)} = \mathbf{H(x)U} \tag{11}$$

where $\mathbf{H}$ is the set of interpolation functions, $\mathbf{x}$ is a vector of all the data points and $\mathbf{U}$ is the vector of displacement components at each feature point. The strains produced at each feature point due to the displacement are obtained as a combination of the element strains associated with the feature points:

$$\epsilon(\mathbf{x}) = \mathbf{B(x)U} \tag{12}$$

where $\mathbf{B}$ is the strain matrix and ϵ is a vector of strains produced at the point under consideration. The problem of modal displacements is then solved as a dynamic equilibrium equation:

$$\mathbf{M\ddot{U} + D\dot{U} + KU = R} \tag{13}$$

where $\mathbf{M}$, $\mathbf{D}$ and $\mathbf{K}$ are the mass, damping and stiffness matrices and $\mathbf{R}$ is the load matrix. The reader is directed to Ref. 10 for detailed derivation of all the mentioned matrices. The non-rigid deformations are then expressed in an orthogonal system where the basis is defined as the set of orthonormalized eigenvectors of $\mathbf{M^{-1}K}$. Given that $\mathbf{x}$ is the set of all feature points, the locations of the new feature points is given as follows:

$$\mathbf{x'} = \bar{\mathbf{x}} + \sum_{\mathbf{i=1}}^{\mathbf{m}} \phi_\mathbf{j} \tilde{\mathbf{u}}_\mathbf{j} \tag{14}$$

where $\bar{\mathbf{x}}$ is the mean displacement position, $\mathbf{x'}$ is the deformed position, $\tilde{u}_j$ is the jth mode parameter value and ϕ_j is the jth eigenvector defining the jth modal displacement. The system can be re-orthogonalized to separate the affine parameters from the modal parameters.

In our formulation, we use an FEM based technique called *active blobs*[29] as a tool to register prototype images. Initial blob of a reference object, I_{ref} is created by associating a deformable polygonal mesh with the object texture map. Registration of a novel image, I_1, is then solved as an energy minimization problem where the shape parameters (in our case, the finite element modes) are estimated so that difference between the warped reference object and the novel object is minimized by least squares approach. The energy minimization problem is formulated as follows:

$$E_{image} = \frac{1}{n}\sum_{i=1}^{n} e_i^2 \tag{15}$$

$$e_i = \|I_1^{'}(x_i, y_i) - I_{ref}(x_i, y_i)\| \tag{16}$$

where $I_1^{'}(x_i, y_i)$ is the intensity of the pixel at location (x_i, y_i) in the inverse warped target image I_1 and $I_{ref}(x_i, y_i)$ is the intensity of the pixel at the same location in the reference image. The adverse effect of the outliers that tend to

throw the minimization process out of track are handled by using a robust error norm which is a *Lorentzian influence function* ρ, given as:

$$E_{image} = \frac{1}{n}\sum_{i=1}^{n}\rho(e_i,\sigma) \tag{17}$$

$$\rho(e_i,\sigma) = \log(1+\frac{e_i^2}{2\sigma^2}) \tag{18}$$

where σ is an optional scale parameter.

5. IMPLEMENTATION STEPS

We have divided the implementation into three stages, namely *average image computation*, *training phase* and *matching phase*. We use Levenberg-Marquardt method[24] for minimization during the matching phase. Below, we provide a brief description of each step. The readers are directed to Ref. 30 for detailed description of the algorithm.

5.1. Average Image Computation

It may be the case that some prototypes are more similar and hence may form clusters in the prototype subspace. If the reference image happens to be selected from one such cluster, then it may not register well with prototypes from other clusters. This phenomenon is called *true shape vulnerability*.[31] In order to avoid this problem we use the average image, which will be fairly equidistant from all prototypes, for registration. This average image is computed in an iterative fashion. We start with an arbitrary reference image I_{ref}. The user circles out the region of interest from which a blob is created. This blob is then registered with the remaining prototypes. A new reference blob is created by averaging the shape and the texture parameters of the prototypes, obtained by the process of registration. This process is repeated to obtain new reference blobs, until the difference between the new reference blob and the old reference blob drops below a threshold.

5.2. Training Phase

Once the *reference* image has been computed, the system is trained by registering all the prototypes with the reference blob. In our implementation, the mode values required for deforming the reference blob to match the prototype are stored as the shape vectors and the inverse warped prototype images are stored as the texture vectors.

5.3. Matching Phase

Matching of a novel image is done by the minimization of the objective function Equation (6). The minimization is performed by Levenberg-Marquardt method. The first and second derivatives of the objective function, required for the minimization process are computed according to the first approximation principle for derivatives. For simplicity, we use forward warping instead of inverse warping. We employ a further constraint on the shape coefficients such that they sum to 1 in order to address redundancy due to the modeling of the affine parameters in the FEM model. The equations for the objective function along with its required derivatives are provided below:

$$E(\mathbf{b},\mathbf{c}) = \frac{1}{2}\sum_{x,y}[I_{novel}(x,y) - \mathcal{W}(\mathbf{b}\cdot\mathbf{T},\mathbf{c}\cdot\mathbf{S})(\mathbf{x},\mathbf{y})]^2 + \gamma(\sum_{\mathbf{k=1}}^{\mathbf{N}}\mathbf{c_k}-\mathbf{1})^2 \tag{19}$$

$$\frac{\partial E}{\partial b_k} = \sum_{x,y}[I_{novel}(x,y) - \mathcal{W}(\mathbf{b}\cdot\mathbf{T},\mathbf{c}\cdot\mathbf{S})(\mathbf{x},\mathbf{y})][-\mathcal{W}(\mathbf{T_k},\mathbf{c}\cdot\mathbf{S})(\mathbf{x},\mathbf{y})] \tag{20}$$

$$\frac{\partial E}{\partial c_k} = \sum_{x,y}[I_{novel}(x,y) - \mathcal{W}(\mathbf{b}\cdot\mathbf{T},\mathbf{c}\cdot\mathbf{S})(\mathbf{x},\mathbf{y})][\frac{\partial\mathcal{W}(\mathbf{b}\cdot\mathbf{T},\mathbf{c}\cdot\mathbf{S})}{\partial\mathbf{c_k}}(\mathbf{x},\mathbf{y})] + \mathbf{2}\gamma(\sum_{\mathbf{k=1}}^{\mathbf{N}}\mathbf{c_k}-\mathbf{1}) \tag{21}$$

$$\frac{\partial\mathcal{W}(\mathbf{b}\cdot\mathbf{T},\mathbf{c}\cdot\mathbf{S})}{\partial c_k}(x,y) = [\mathcal{W}(\mathbf{b}\cdot\mathbf{T},(\mathbf{c}+\mathbf{\Delta c})\cdot\mathbf{S})(\mathbf{x},\mathbf{y}) - \mathcal{W}(\mathbf{b}\cdot\mathbf{T},\mathbf{c}\cdot\mathbf{S})(\mathbf{x},\mathbf{y})]/\mathbf{\Delta} \tag{22}$$

$$\Delta c_k = [0,\ldots,k-1 times,\Delta,0,\ldots,0] \tag{23}$$

The second derivatives of the given function are approximated as below:

$$\frac{\partial^2 E}{\partial m_i \partial m_j} = \frac{\partial E}{\partial m_i}\frac{\partial E}{\partial m_j} \tag{24}$$

where $m_k = c_k$ or b_k. These derivatives are then substituted into Equations 9 and 10 to compute the gradient vector and the approximate Hessian matrix required in the Levenberg-Marquardt method. The parameter vector q is defined as a composite vector of the shape and the appearance parameters:

$$q = [\mathbf{c}|\mathbf{b}] \tag{25}$$

q is updated by the change in the parameter vector, Δq estimated by Equation 7. This process is iterated until the final error magnitude drops below a given threshold or a fixed number of iterations are completed. The λ in Equation 7, acts as a time-varying control parameter that forces the solution to follow the steepest gradient in order to converge to the minimum.

6. RESULTS

The system was tested with two types of datasets, namely face images and sequences of heart images, and was tested with some novel images that were not present in the training set. The main points for which we tested the system are following:

- the algorithm should be able to reconstruct novel images by appropriately combining prototype images.
- the algorithm should be capable of handling significant variations (e.g. "gender", in our experiments for the face images).
- test the robustness of linear combinations paradigm to reconstruct novel images, that belong to the same object class but have not been seen in the training set (e.g. the algorithm can reconstruct images of men without mustaches, by appropriately combining images of women and men with mustaches).
- generalization of the technique to objects other than faces (e.g. heart images, in our experiments).

6.1. Test Set 1: Face Images

The code for registration of images was taken from *Active Blobs* which is available on the internet[†]. The prototype set comprises of random face images drawn from the MIT database[‡] (see Figure 2). Several novel face images, which were not present in the training set, were tested. All of those could be reconstructed in the combination of parameters paradigm described earlier. The images are of dimension 128x128 pixels. The size of the faces inside the images was typically around 64x64 pixels. The implementation makes extensive use of the graphics hardware for texture mapping and bilinear interpolation. Currently the reconstruction of a novel face image takes around 8 minutes on a R5000 SGI O2, 180 MHz machine. Majority of time is spent in combining the prototype images at each iteration for the reconstruction of novel image. The texture vectors for the prototype images comprise of the whole texture. Significant speedup is possible by dimensionality reduction. In future, we intend to evaluate the system with *dimensionally reduced* prototype texture vectors, where we will use coefficients obtained by projecting prototype images into the eigenspace instead of textures. We expect the performance of the system would improve as we will have to combine less number of eigen-images for reconstruction. We can further reduce the computation time by doing minimization on only one color channel. The average image for the dataset is given in Figure 3(a) and some results of matching of novel images have been provided in Figures 3(b), (c), (d), (e) and (f).

[†]http://www.cs.bu.edu/groups/ivc/

[‡]ftp://whitechapel.media.mit.edu/pub/images/

Figure 2. Prototype face images (#prototypes = 100). This training set comprises of 75 images of males with mustaches and 25 images of females.

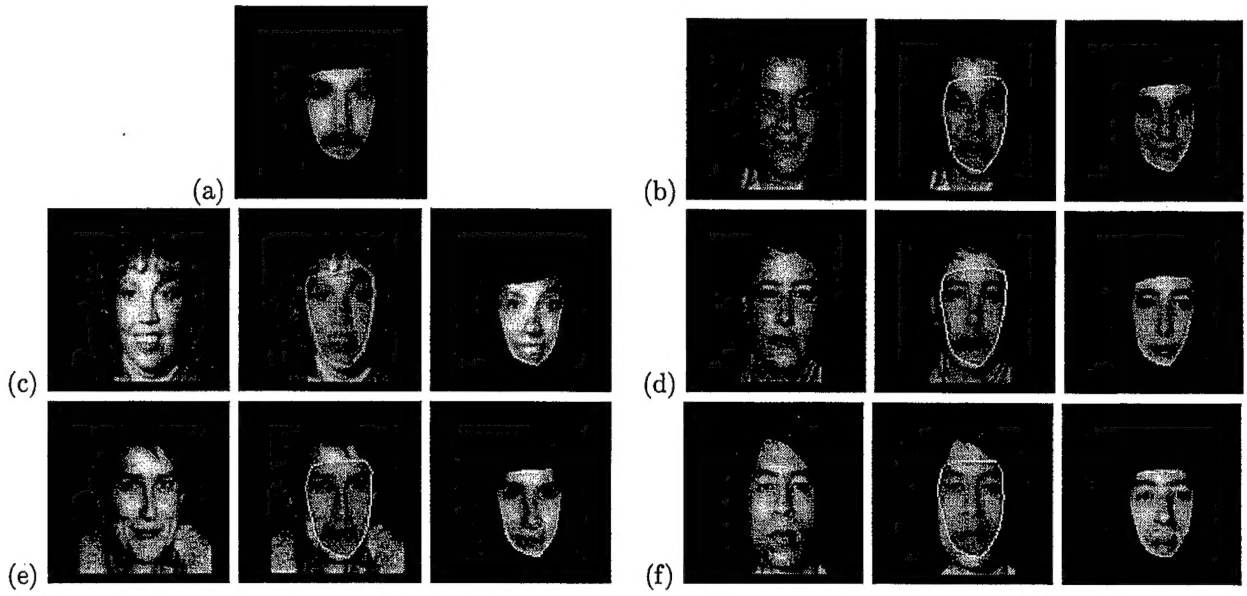

Figure 3. Linear combination of face images: (a) Average face image; (b), (c), (d), (e), (f) Reconstructed novel face images obtained by the combination of shape and texture parameters of the prototype face images (Left: input novel image; Middle: average image registration; Right: reconstruction of the circled region).

6.2. Test Set 2: Sequences of Heart Images

We tested the system on images of heart taken from the MIT heart database[§], in order to evaluate the generality of the approach. Since there were only 38 images, we included all the odd numbered images in the training set and used the even numbered images as novel images. The images used for training are given in Figure 4(a). The average image for this sequence of images and reconstruction of some novel images are given in Figures 4(b) and 4(c), (d), (e) and (f). As may be seen, the approach was able to reliably reconstruct various intermediate stages of heart pumping. The estimated shape and texture parameters, obtained from the reconstruction, can be used for various medical applications.

7. DISCUSSION

Levenberg-Marquardt method is a quadratic minimization technique, that requires significant amount of time for computation of the Hessian matrix at each step. The major bottleneck is the number of floating point multiplications involved which is $O(n^2m^2)$ where each image has $O(n^2)$ pixels and there are $O(m)$ prototype images. This issue needs to be addressed in order to make the matching process real-time. Currently we are exploring different heuristics for speedup. These heuristics are primarily focussed on reducing the net computations and the actual number of prototypes to combine.

A possible approach to significantly reduce the computation time at each step, is to compute the Hessian matrix over random pixels, rather than over complete images. This, intuitionally, simulates the *stochastic gradient method*,[24] but would be more efficient as it would converge to the solution faster by taking larger step sizes (implicitly controlled by λ) at each iteration, provided the error function happens to be quadratic.

The number of computations involved for minimization is quadratically related to the number of prototypes. Hence, selection of an optimal set of prototypes is paramount to reducing the implementation time. Currently, a set of prototypes is chosen randomly and the training is done on this set. If three prototypes are thought to lie on a line in the prototype space, then any number of extra prototypes on the same line are redundant and hence should be singled out. Though different statistical methods like k-means clustering, hierarchical clustering, Bayes classifier etc. can be used, a normal tradeoff involved is that typical pattern recognition methods require large training data sets

[§]ftp://whitechapel.media.mit.edu/pub/images/

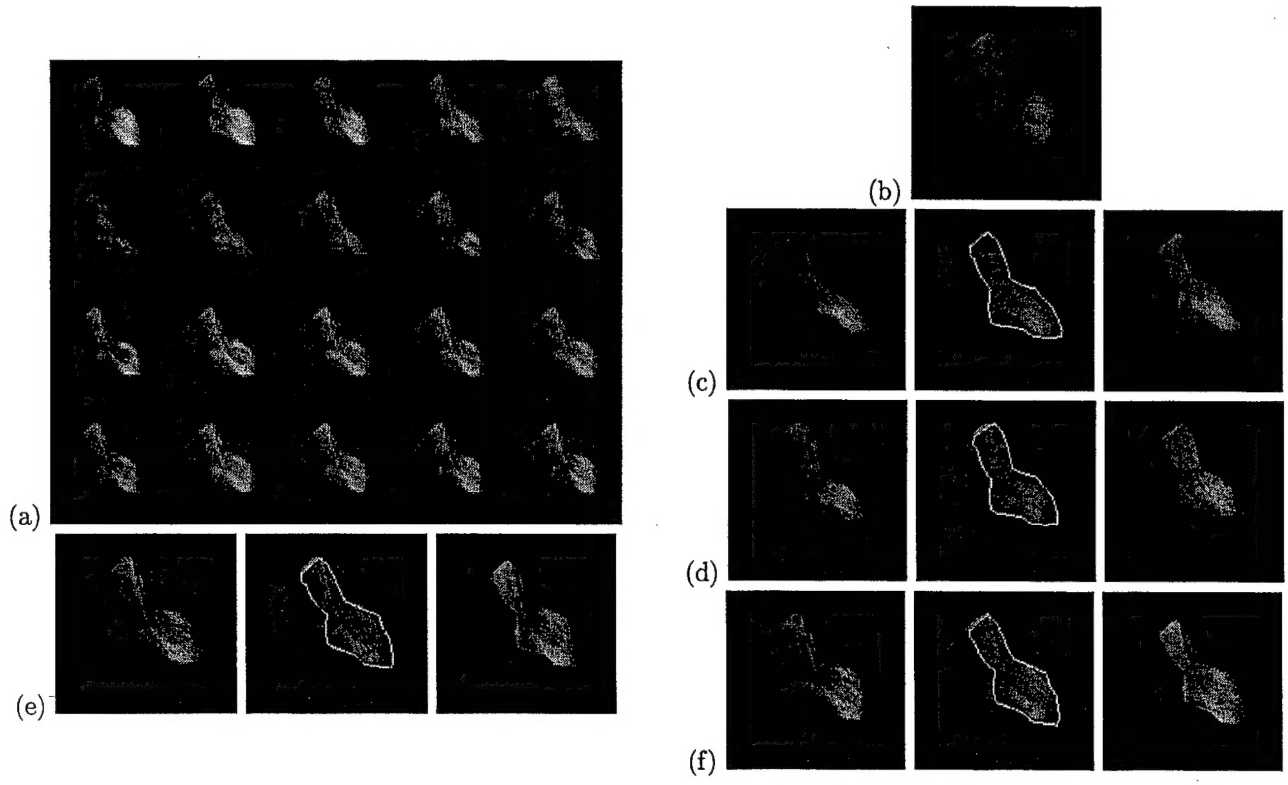

Figure 4. Linear combination of heart images: (a) Prototype heart images (#prototypes = 20. This training set comprises of various intermediate images of contraction and expansion of the human heart while pumping blood); (b) Average heart image; (c), (d), (e), (f) Reconstructed novel heart images obtained by combining the shape and texture parameters of the prototype heart images (Left: input novel image; Middle: average image registration; Right: reconstruction of the circled region).

which are diverse enough to characterize the whole object class.[27,32] We are currently exploring different techniques to be able to select sets of "good" prototypes in future.

8. CONCLUSION

We presented a model-based linear combinations approach for modeling objects. The methodology, implementation status, results obtained so far and possible explanations of various observed behavior have been described. Apart from these, the method was compared with existing active appearance model and the pros and cons were brought out. Also various ways of extending the existing framework have also been described. In future, we plan to implement clustering algorithms for appropriately choosing the representative set of prototypes. Apart from this, we intend to study the application of the given approach for various computer vision problems viz. recognition, image registration and analysis, image compression and morphing.

ACKNOWLEDGMENTS

This work was supported in part through Office of Naval Research Young Investigator Award N00014-96-1-0661, and National Science Foundation grants IIS-9624168 and EIA-9623865.

REFERENCES

1. S. Y. Edelman and H. H. Bulthoff, "Viewpoint-specific representations in three dimensional object recognition," Tech. Rep. A.I.Memo 1239, MIT, 1990.

2. P. Sinha, *Perceiving and recognizing 3D forms.* PhD thesis, Massacheussetts Institute of Technology, 1995.
3. N. Logothetis, J. Pauls, and T. Poggio, "Viewer-centered object recognition in monkeys," Tech. Rep. AI Memo No. 1473, CBCL Paper No. 95, M.I.T. AI Lab. and CBIP Whitaker College, April 1994.
4. N. Logothetis, J. Pauls, and T. Poggio, "Shape representation in the inferior temporal cortex of monkeys," *Current Biology* **5**(5), pp. 552–563, 1995.
5. H. H. Bulthoff, S. Y. Edelman, and M. J. Tarr, "How are three-dimensional objects represented in the brain?," *Cerebral Cortex* **5**(3), pp. 247–260, 1995.
6. S. Nayar and T. Poggio, eds., *Early Visual Learning*, Oxford University Press, 1996.
7. M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International Journal of Computer Vision* **1**(4), pp. 321–331, 1988.
8. T. F. Cootes, D. H. Cooper, C. J. Taylor, and J. Graham, "Trainable method of parametric shape description," *Image and Vision Computing* **10**, pp. 289–294, June 1992.
9. A. Pentland and S. Sclaroff, "Closed-form solutions for physically based modeling and recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **13**, pp. 715–729, July 1991.
10. S. Sclaroff and A. Pentland, "Modal matching for correspondence and recognition," Tech. Rep. 201, M.I.T. Media Laboratory Perceptual Computing Section, May 1993.
11. T. F. Cootes and C. J. Taylor, "Combining point distribution models with shape models based on finite element analysis," *Image and Vision Computing* **13**, pp. 403–409, June 1995.
12. M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience* **3**(1), pp. 72–86, 1991.
13. A. Pentland, B. Moghaddam, and T. Starner, "View-based and modular eigenspaces for face recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 1994.
14. H. Murase and S. K. Nayar, "Visual learning and recognition of 3-d objects from appearance," *International Journal of Computer Vision* **15**, pp. 5–24, 1995.
15. B. Moghaddam, W. Wahid, and A. Pentland, "Beyond eigenfaces: Probabilistic matching for face recognition," in *Proceedings of the Third International conference on Automatic Face and Gesture Recognition*, April 1998.
16. A. Lanitis, C. J. Taylor, and T. F. Cootes, "Automatic face identification system using flexible appearance models," *Image and Video Computing* **13**, pp. 393–402, June 1995.
17. C. Nastar, B. Moghaddam, and A. Pentland, "Generalized image matching: Statistical learning of physically-based deformations," in *ECCV96*, (Cambridge, England), April 1996.
18. S. Ullman and R. Basri, "Recognition by linear combinations of models," *IEEE Transactions on Pattern Recognition and Machine Intelligence* **13**, October 1991.
19. M. Jones and T. Poggio, "Model-based matching of line drawings by linear combinations of prototypes," Tech. Rep. AI Memo No. 1559, CBIP Paper No. 128, M.I.T. AI Lab. and CBIP Whitaker College, December 1995.
20. M. Jones and T. Poggio, "Model-based matching by linear combinations of prototypes," Tech. Rep. AI Memo No. 1583, CBIP Paper No. 139, M.I.T. AI Lab. and CBIP Whitaker College, November 1996.
21. M. Jones and T. Poggio, "Multidimensional morphable models," in *International Conference of Computer Vision*, (Bombay, India), January 1998.
22. T. Vetter and T. Poggio, "Linear object classes and image synthesis from a single example image," Tech. Rep. AI Memo No. 1531, CBIP Paper No. 119, M.I.T. AI Lab. and CBIP Whitaker College, March 1995.
23. T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active appearance models," in *ECCV98*, 1998.
24. S. Teukolsky, W. Press, B. Flannery, and W. Vetterling, *Numerical Recipes in C*, Cambridge University Press, UK, 1988.
25. D. Terzopoulos, "Image analysis using multigrid relaxation methods," *IEEE Transactions on Pattern Recognition and Machine Intelligence* **8**(2), pp. 129–139, 1986.
26. A. Rosenfeld, ed., *Multiresolution Image Processing and Analysis*, Springer-Verlag, New York, 1984.
27. S. Sclaroff, "Deformable prototypes for encoding shape categories in image databases," *Pattern Recognition* **30**, April 1997.
28. K. Bathe, *Finite Element Procedures in Engineering Analysis*, Prentice Hall, 1982.
29. S. Sclaroff and J. Isidoro, "Active blobs," in *International Conference of Computer Vision*, (Bombay, India), 1998.
30. S. Sethi and S. Sclaroff, "Combinations of deformable shape prototypes," Tech. Rep. 99-007, Computer Science Department, Boston University, July 1999.

31. A. Hill and C. J. Taylor, "Automatic landmark generation for point distribution models," in *Proceedings of the 5th British Medical Vision Conference*, pp. 429–438, 1994.
32. R. O. Duda and P. E. Hart, *Pattern Recognition and Scene Analysis*, John Wiley, New York, 1973.

REPORT DOCUMENTATION PAGE

OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	August 1999	

4. TITLE AND SUBTITLE

Combinations of Non-Rigid Deformable Appearance Models

5. FUNDING NUMBERS

G N00014-96-1-0661

6. AUTHOR(S)

Saratendu Sethi and Stan Sclaroff

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Computer Science Department
Boston University
111 Cummington Street
Boston, MA 02215

8. PERFORMING ORGANIZATION REPORT NUMBER

sclaroff-ONR-TR99-321

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Department of the Navy
Office of Naval Research
Ballston Centre Tower One
800 North Quincy Street
Arlington, VA 22217-5660

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

A Framework for object recognition via combinations of nonrigid deformable appearance models is described. An object category is represented as a combination of deformed prototypical images. An object in an image can be represented in terms of its geometry (shape) and its texture (visual appearance). We employ finite element based methods to represent the shape deformations more reliably and automatically register the object images by warping them onto the underlying finite element mesh for each prototype shape. Vectors of objects from the same class (like faces) can be thought to define an object subspace. Assuming that we have enough prototype images that encompass major variations inside the class, we can span the complete object subspace. Thereafter, by virtue of our subspace assumption, we can express any novel object from the same class as a combination of the prototype vectors. We present experimental results to evaluate this strategy and finally, explore the usefulness of the combination parameters for analysis, recognition and low-dimensional object encoding.

14. SUBJECT TERMS

Non-rigid shape description and recognition; image and video database indexing; content based retrieval

15. NUMBER OF PAGES

12

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
unclassified	unclassified	unclassified	UL

ENCL (1).